The Bell System Technical Journal
Vol. 53, No. 1, January 1974
Printed in U.S.A.

Automatic Intercept System:

Operational Programs

By H. COHEN, D. E. CONFALONE, B. D. WAGNER, and W. W. WOOD

(Manuscript received May 16, 1973)

The Automatic Intercept System Operational Programs provide the logic for processing calls served by the system. These programs also perform administrative and software correction and recovery functions. Described are program organization, use of temporary memory, and details of call processing.

I. INTRODUCTION

The stored program for the Automatic Intercept System (AIS)[1] directs the operation of equipment which processes intercept calls. The system is designed to meet stringent operational requirements similar to those imposed on other stored program switching systems, such as the No. 1 Electronic Switching System (ESS).[2] AIS provides rapid, reliable, and economical intercept service and supports simple administrative and maintenance procedures.

This paper describes the programs that process calls as well as some of the administrative and maintenance programs. Companion papers cover the remaining parts of the AIS program.[3–5]

Intercept calls differ in several significant ways from calls served by typical local or toll switching centers. These differences have been exploited wherever possible to simplify the system design. Most influential among these factors are the following:

(*i*) The variety of services and number of options required are relatively limited.
(*ii*) There are no subscriber-to-subscriber connections.
(*iii*) Calls have short holding times (the average call is less than

30 seconds in duration, while over 90 percent of the calls to AIS last less than one minute).

(*iv*) Connections to callers receiving announcements are changed at a high rate (every 0.5 or 1.5 seconds).

(*v*) Several hundred thousand intercepted lines (in an area of several million lines) can be served by an AIS.

These factors have most influenced the system design in three areas: use of a standard generic program, novel methods of storing and administering installation-dependent data, and wide use of autonomous circuits. These areas are considered in the following paragraphs.

1.1 Generic program

Intercept service is much more limited in scope than the service provided by a typical telephone switching office. Thus, a single *generic* program can serve all sites. This approach is efficient because the small number of options do not significantly penalize installations where certain options are not used. Use of a single generic program is also desirable because it simplifies testing and support of programs and minimizes the possibility of undiscovered incompatibilities.

1.2 Installation-dependent data

Installation-dependent data consist of:

(*i*) intercept number records, and
(*ii*) records describing installation options.

The intercept number records are stored on disc files.[5] These memory devices provide an economical fast storage medium for the large volume of data required. These data do not describe the way an installation is equipped, but merely describe intercepted lines in local offices served by an AIS. Thus, such records are not considered installation options.

The remaining installation-dependent data, known as *nongeneric data*, describe the way a particular AIS is equipped and operates. Such data include trunk and announcement translation information, equipment and feature options installed, etc. Backup records of nongeneric data are stored on the disc files, but the working record is kept in the temporary memory, *call store*, for rapid access by the operational programs.

1.3 Autonomous circuits

In order to minimize repetitive processing operations and maximize system call capacity, a high degree of autonomous circuit operation is

provided. Among the circuits with some autonomous operation are:

(*i*) a time-division switching network, which allows simple path selection and network control and expedites frequent connection changes;[3]
(*ii*) control unit circuits which perform digit receiving and data sending;
(*iii*) associative match circuitry in the disc file control for intercept number lookup;
(*iv*) a trunk scanner associated with the time-division switching network; and
(*v*) multifrequency receiver trunk circuits which perform supervisory wink signaling.

II. PROGRAM ORGANIZATION

The AIS program operates in real time to process intercept calls. The program is structured to perform each task at an appropriate rate so that response times to input stimuli are minimized and processing capability is maximized.

Parts of the AIS generic program were developed for use by both AIS and the No. 2 Electronic Switching System.[6] Since these systems use the same control complex, programs which perform functions such as control unit maintenance or teletypewriter input-output are substantially application-independent and thus can be used in common by both systems. The remaining programs, used solely for the AIS application, include all call processing and administrative programs as well as application-dependent maintenance programs. The total program consists of almost 100,000 program store words. Figure 1 illustrates the breakdown between commonly used and application programs as well as the functional division.

In addition to certain operational programs, several *service programs* are shared by the AIS and No. 2 ESS developments.[7] These progams, written for a general-purpose computer, are used to prepare and test operational programs and to produce program documentation. Among the service programs are a macro assembler, a linking loader, and a control unit simulator.

2.1 Main program loop

The *main program loop* is a series of programs executed in a repeating cycle at *base level*, that is, when no interrupts are in effect. Programs in this loop (Fig. 2) perform five functions at the same frequency: call processing, maintenance, teletypewriter, call store audits, and file

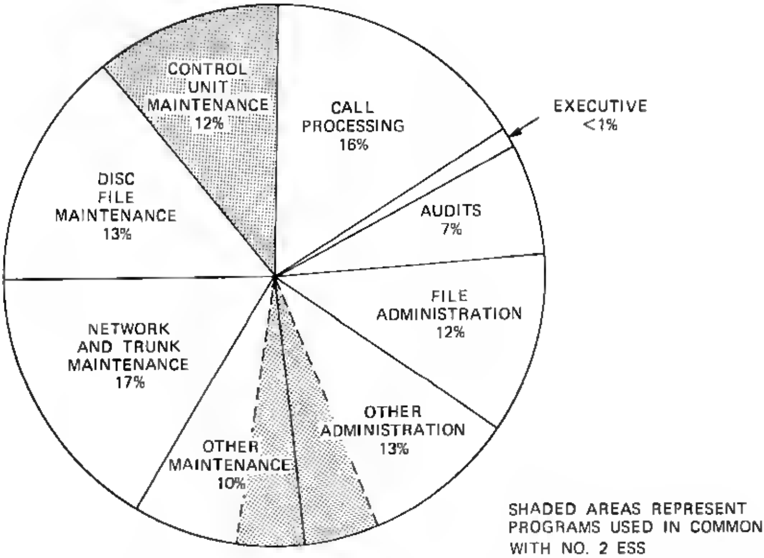


Fig. 1—Functional division of AIS programs.

administration. Following is a description of these programs and their control.

2.1.1 Executive control program

Sequencing of base level programs is performed by the *executive control* program. Programs which are directly subordinate to executive control are known as *monitor* programs. Each monitor program normally is entered once per main program loop. However, the executive may bypass one or more monitors during a particular main program loop based on indicators of certain abnormal system activity. These abnormal modes of operation include:

(*i*) reloading of nongeneric data from a tape backup (Section 7.2).* In this mode, call processing activity is suspended and the call processing monitor is not entered and

(*ii*) system initialization, when only call store audit (Section 5.4) and file administration[5] programs are entered to initialize nongeneric call store data.

* Note that should disc file intercept number records be lost, they are reloaded from a backup record. During this period, call processing continues and system programs are executed normally.

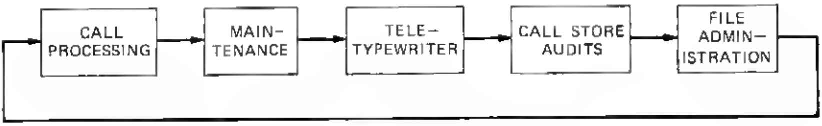


Fig. 2.—Main program loop.

Additionally, the executive schedules programs for execution based on time of day and elapsed time. To perform these services, the executive maintains a highly accurate time-of-day clock in call store memory.

2.1.2 Monitor programs

The call processing monitor (Section IV) responds to requests for service detected by interrupt level programs. It also controls the execution of client task programs which advance calls through various stages. The teletypewriter monitor controls the receiving and sending of data over the six AIS teletypewriter channels. Character input and output functions are performed at an interrupt level. The call store audit monitor (Section V) controls execution of audit programs. The file administration monitor[5] controls programs which perform disc file access and which administer the intercepted number data base.

The maintenance monitor controls all system maintenance activity. Based on the maintenance states of system equipment and requests for maintenance action, this monitor selects for execution those maintenance jobs immediately vital to continued operation of the system. Other functions of the maintenance monitor include:

(*i*) reacting to changes of status of the control units;
(*ii*) scheduling maintenance action on all system equipment;
(*iii*) responding to maintenance and utility requests input manually via a maintenance teletypewriter or the maintenance center control panel;
(*iv*) testing for and reacting to abnormal conditions indicated by the states of alarm-indicating master scanner ferrods; and
(*v*) initiating recovery action after detecting that the main program loop is not cycling properly.

The scheduling function of the maintenance monitor is complex and relies on a preselected set of priorities associated with each class of maintenance test actions. The relative priorities of the classes have been determined by considering the consequences of deferring each class. For example, diagnostic testing of a control unit resulting from

an error detected by a check circuit is given priority over daily routine testing of the switching network.

The maintenance scheduler is entered once per main program loop. During each entry, the scheduler permits a portion of a maintenance test to be executed. Maintenance tests are segmented since, in general, they would require hundreds of milliseconds if executed continuously. The maintenance scheduler generally permits only a single maintenance function to be executed until it is completed to prevent interference between maintenance programs which may temporarily leave equipment in an arbitrary state.

Flexibility is afforded in maintenance scheduling by prematurely terminating, or *aborting*, a maintenance function which may be in progress when a higher priority function is requested. An aborted function may be resumed when all higher priority functions have been completed; also, a new request is held until all higher priority requests have been honored. In addition, the scheduler can disallow particular functions when indicators do not permit that function to be executed. For example, when the control units are operating in synchronism, diagnostic testing of the offline control unit is not allowed.

Manual requests via teletypewriter for maintenance tests permit the requestor to specify that the test be performed in one of three modes: once, repetitively, or in *step mode* (once on each operation of a key from a manual remote test facility). In repetitive or step modes, the maintenance scheduler controls the repeated execution of the test and can cause the result of the last test (pass or fail) to be indicated on lamps at the maintenance center display panel and the remote test facility. These features are most helpful in repairing faulty equipment units.[3,4]

Some programs which are subordinate to the maintenance monitor are known as subsystem maintenance monitor programs and are covered in companion papers.[3,4] Maintenance of the No. 2 ESS control complex is also covered elsewhere.[8]

2.2 *Interrupt structure*

Multilevel interrupt facilities are provided in the control unit to permit entry to programs immediately on demand. Interrupts have a priority structure such that the highest interrupt level demanded is executed. When an interrupt signal is generated, a program is entered at a fixed address corresponding to that interrupt level. When the interrupt is completed, control is returned to the program that was interrupted.

Table I — Interrupt structure

Interrupt	Source	Purpose(s)
High-Priority Maintenance	Demand by control unit	(*i*) Control unit mismatch recovery (*ii*) Digit scanning error recovery (*iii*) Utility request processing
Disc File 0 Disc File 1	Demand by disc file	Disc file block transfer input–output
Input-Output 25-Millisecond	Timed by control unit	Other input–output
Low-Priority Maintenance	Demand by control unit	Continue control unit mismatch recovery

Three types of interrupts are used in AIS, and tbey are assigned at five priority levels. A list of interrupts showing tbeir relative priorities is contained in Table I.

2.2.1 25-millisecond interrupt

A periodic 25-millisecond interrupt, IO25, is generated by circuits within the control unit. This interrupt handles:

(*i*) Many input-output functions requiring execution more frequently than once per main program loop. Among tbese tasks are detection of trunk supervisory changes, processing of multifrequency digits, and disc file lookups (see Section IV).

(*ii*) Tasks requiring precise timing between entries, sucb as teletypewriter input-output, or maintenance testing the duration of timing signals provided by peripberal circuits where tbe signal may last hundreds of milliseconds.

Tbe IO25 interrupt programs are sequenced by a portion of the executive control operating at IO25 interrupt level. Tbe control structure used in this interrupt is kept simple, since inefficiencies here would be multiplied by the high frequency of execution and thus would be wasteful of processing time. Hence, task programs generally interface directly with the IO25 executive.

The IO25 executive controls execution in one of four modes. Normally, all task programs are executed, but in other (abnormal) modes of operation, only essential tasks are executed. Tbese other modes are:

(*i*) recovery from control unit mismatcb;
(*ii*) reloading of nongeneric data from paper tape backup; and
(*iii*) system initialization.

Table II contains a list of IO25 interrupt functions in their order of execution and indicates the modes in which each is executed.

The duration of the IO25 interrupt is normally much less than 25 milliseconds. Should an occasional interrupt last longer than 25 milliseconds, "jitter" could he introduced in the timing of precisely timed tasks.

2.2.2 Disc file interrupts

Two *disc file interrupts* are provided, one per disc file, at adjacent priority levels. These interrupts are used to transfer large blocks of data between a control unit and a disc file (other than for routine call lookups) hecause the data cannot all be buffered in file control registers.[4] Data transfers requiring file interrupts are used for:

(*i*) administration and auditing of the intercept numher data base[5] and the hackup nongeneric data;
(*ii*) auditing of nongeneric data in call store; and
(*iii*) maintenance of the disc file suhsystem.

Further details of the disc file interrupts are discussed in a companion paper.[5]

2.2.3 Maintenance interrupts

Two maintenance interrupts are used to process certain maintenance and manual requests. The *high-priority maintenance interrupt*, the highest-priority system interrupt, is used for:

(*i*) processing of manual utility requests from the maintenance center;
(*ii*) recovery from errors occurring in autonomous multifrequency digit scanning performed hy the control unit; and
(*iii*) recovery from mismatches of the control units.

Processing requests for (*i*) and (*ii*) is rapid and, therefore, can he completed at this interrupt level. However, in case (*iii*), testing which ensues after a control unit mismatch requires up to 220 milliseconds. If all this testing were performed in this interrupt, all other system programs, including lower-priority interrupts, would not be executed during this period; this would most likely result in mishandling of some calls. Consequently, after several milliseconds at this interrupt level, the remaining testing is performed at *low-priority maintenance interrupt* level, the lowest-priority system interrupt. During this time, only hase level work is delayed, permitting call processing input-output tasks to continue.

Table II — IO25 interrupt functions

Description	Mode of Operation			
	Normal	CU Mismatch Recovery	Tape Mode	System Initialization
Update system time	X	X	X	X
High-precision timing for teletypewriter maintenance	X		X	
Data outpulsing	X	X		
Teletypewriter input/output processing	X		X	
Autonomous scanner processing	X			
Announcement connections	X	X		
Disc file input/output	X		X	X
Digit receiving	X	X		
Service observing	X	X		
Directed scan processing	X			
Low-precision timing for announcement machine maintenance	X		X	

III. USE OF CALL STORE MEMORY

The call store in the AIS control complex is a direct-access memory consisting of modules of ferrite sheets operating on a 6-microsecond read-change-write cycle.[6] Each module is comprised of 4096, 16-bit words. Areas used by the operational programs may be classified into five categories:

(*i*) records of calls currently in progress;
(*ii*) maps which record the busy, idle, or maintenance states of peripheral equipment, trunks, and service circuits;
(*iii*) records of nongeneric data particular to an Automatic Intercept Center (AIC);
(*iv*) buffers, hoppers, and registers which receive (and transmit) data from (to) peripheral equipment; and
(*v*) traffic and plant administration and call queuing data.

Since the overall AIS call store requirements are small (compared with other stored program switching systems such as No. 1 ESS or

No. 2 ESS), enough call store is reserved at every installation to meet the maximum call capacity of an AIC. If a particular AIC is not fully equipped, the call store words which would have been associated with the nonexistent facilities are simply not used. This dedication of all call store areas eliminates any need for redefining call store layouts to accommodate changes in an operating AIC; thus, it is completely compatible with the generic program concept.

3.1 Classes of call store data

Data stored in the call store memory can be classified into one of three categories according to their use and frequency of alteration: transient data, semipermanent data, and stable data.

3.1.1 Transient data

Transient data words are changed frequently and are associated with the processing of a particular call or maintenance action. For example, the digits received by the AIC from a local office, and the linkages between call-active equipment control registers, are transient data.

3.1.2 Semipermanent data

Semipermanent data call store words contain the nongeneric data associated with a particular AIC. The semipermanent data in call store is backed up on both disc files and is altered only by plant changes (Section 7.2). An example of nongeneric data is the information describing the equipping of a trunk network.

Certain call store words contain both transient and semipermanent data. This permits more efficient call processing, but increases the complexity of initializing and auditing these words. Many of the equipment control registers (Section 3.3) are examples of shared data type words.

3.1.3 Stable data

Stable data words contain information which is retained by the system over an extended period of time, which may be updated periodically, and which, if lost, is very difficult to reconstruct. They are similar to transient data words, except that they are not related to the major call processing and maintenance functions of the system. Thus, it is unlikely that errors in these words would cause severe software problems; hence, they are never reinitialized by an automatically triggered system initialization (Section VI). The time of day is an

example of stable information. Once entered into the system, it is frequently updated and remains in the system almost indefinitely.

3.2 Call-in-progress register

An eight-word control register, known as a *call-in-progress register* (CIPR) (Fig. 3), is assigned to every call as it "enters" the system. The same CIPR stays with the call until it is disconnected. It is used to maintain an up-to-date record of the state of the call and controls the processing of the call in the system.

There are 128 CIPRs per switching network. Since each switching network can connect 64 calls simultaneously, approximately 64 additional calls can enter the system in various queues before the AIS begins to reject originations from the local offices (Section 4.2).

Word 0 of each active CIPR contains the state of the associated call, known as a *progress mark*. The state is encoded such that it equals the address of the beginning of the program routine which handles this state. Access to progress marks is facilitated by a special-purpose "macro-type" control unit command that causes a transfer based on the nonzero contents of the first word of the CIPR.[6] If the contents of the word are zero, no call is associated with this CIPR and the command advances to the next CIPR.

Once during every main program loop each CIPR is accessed and control is transferred to the program indicated by a nonzero progress mark. That program then determines if there is any new information

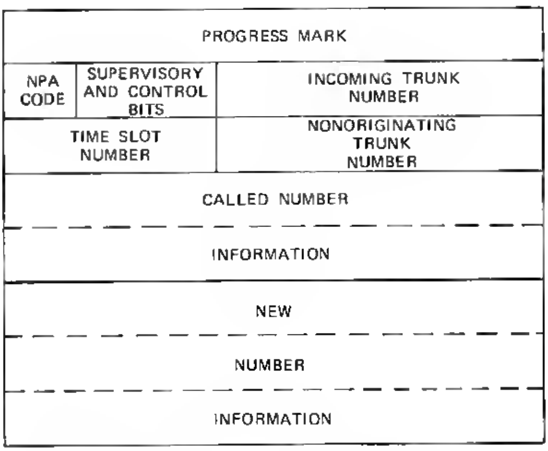


Fig. 3—Call-in-progress register (CIPR) format.

about the call and, hence, if any new action is to he performed on this call. When there is no action to he taken, the progress mark is not changed and the next CIPR is looked at. If there is new information, the program takes the appropriate action; if this action changes the stage of the call, a new progress mark is assigned to the call reflecting the new stage, and the progress mark is written in the CIPR. Tbe action taken on a progress mark entry advances thc call to a point where no further processing is possihle without a real-time break. Typically, the hreak is necessary when waiting for unavailable facilities, for response from a peripheral circuit, or for a timed period to elapse.

The efficiency of this nonzero progress mark approach for all active calls results from two characteristics of intercepted calls:

(*i*) most intercepted calls bave a sbort bolding time, and
(*ii*) active calls in the system are usually not in a stable state (e.g., announcement connections are changed every 0.5 or 1.5 seconds).

Hence, the little processing time wasted monitoring stahle connections is justified by the simplicity achieved in performing the call processing function.

The second and third words of the CIPR are used to store the originating trunk equipment location and the equipment location of the facility to which it is connected, as well as supervisory information for hoth trunks and the number of the network time slot in which the connection is made. The rest of tbe CIPR contains the called number dialed by the customer or keyed by the operator, a digit indicating the call class, and the file reply for calls requiring a disc file lookup.

3.3 Facility control registers

In addition to the controlling CIPR, a time slot and various other hardware facilities are needed at different stages of the call to process it properly. These hardware facilities are known as *nonoriginating equipment* since they are used to help process calls and do not originate work. Wben active on a call, both the incoming trunk and the various items of nonoriginating equipment are linked to the controlling CIPR. To do this a call store facility control register is associated with each piece of equipment and the number of the controlling CIPR is recorded in this register.

These control registers are arranged hy facility type and are generally one or two words long. In addition to recording the numher of the controlling CIPR of an active facility, these facility control registers

CIPR NUMBER	ANNOUNCEMENT SEQUENCE COUNT

Fig. 4—Time slot word format.

may contain semipermanent data pertinent to the piece of equipment involved. All incoming switching network trunks and time slots have associated facility control registers.

3.3.1 Time slot words

The AIC interconnects trunks via a time-division switching network that provides 64 time slots in which simultaneous connections are made.[3] Associated with each time slot is a call store *time slot word* (Fig. 4). When a time slot is in use, the time slot word contains the number of the CIPR controlling the associated call. Additionally, when a call is in the announcement stage, an *announcement phrase count* is kept of the number of 0.5-second time periods which have elapsed since the announcement sequence began. This count is used to determine the next announcement connection to be made for the associated call.

3.3.2 Primary trunk words

Most of the 512 trunks or service circuits (except the 96 announcement trunks) can appear at any equipment location on the switching network. Therefore, a detailed layout of switching network equipment is provided in a contiguous block of call store known as *primary trunk words* (Figs. 5a and b). Primary trunk words are ordered according to the equipment location of the associated trunks (i.e., numbered position on the network). For example, trunk number 106 (group 1, vertical 0, horizontal 6) would have its corresponding call store word in the 106th (octal) slot of the primary trunk block.

(a)

NPA CODE	M	INCOMING TRUNK TYPE NUMBER	CIPR NUMBER OR MAINTENANCE CODE

(b)

N		M	NONORIGINATING TRUNK TYPE NUMBER	HUNT NUMBER

N = NETWORK NUMBER
M = MAINTENANCE CODE

Fig. 5—Primary trunk word format.

Each primary trunk word contains the *type number* of tbe associated trunk. This type number designates the function of tbe trunk or, in some cases, distinguishes trunks with different supervisory arrangements. Each word also contains two bits indicating the present maintenance state of the trunk.

Tbe remaining bits of a primary trunk word vary according to tbe type of trunk. In general, if the word is for an originating trunk it contains the number of the associated CIPR or a code describing a current maintenance condition. For incoming intercept trunks, also included is a code used to indicate from which of four possible NPAs served by the AIC this trunk originates. For trunks and service circuits which are selected, a trunk selection number (or *hunt number*), assigned when the equipment is installed, is recorded in the primary trunk word. For outgoing trunks to Central Intercept Bureau (CIB) operator positions, an indication of tbe network on which the associated incoming trunk from the CIB position appears is also included.

3.3.3 Secondary trunk words

The primary trunk word for a nonoriginating trunk (other than a test trunk and *Touch-Tone*® receiver) provides translation from tbe trunk equipment location to its hunt number. The *secondary trunk word* (Fig. 6) provides the reverse translation, that is, from the hunt number to the trunk equipment location. Each type of huntable trunk and service circuit has an associated block of secondary trunk words.

Each secondary trunk word contains the equipment location of the corresponding trunk. When the trunk is active on a call or placed in a maintenance state, tbe number of the controlling CIPR or a maintenance code specifying the exact condition, respectively, is also stored therein.

In a typical AIS, a small number of operators are needed. Therefore, outgoing trunks to operator positions are multiplied to botb networks. Thus, each operator position in a two-network installation has the same outgoing trunk appearance on each network and only one secondary trunk word.

3.3.4 Facility selection words

Hunted trunks and service circuits, CIPRs, and network time slots are assigned facility selection words known as *hunt words*. Eacb block of these words records the current busy/idle status of facilities of that type. These selection words are divided into categories which depend upon tbe extent to which traffic is to be distributed evenly over that

TRUNK EQUIPMENT LOCATION	CIPR NUMBER OR MAINTENANCE CODE

Fig. 6—Secondary trunk word format.

group of facilities. Usage of outgoing trunks to operator positions and to service assistant consoles is precisely equalized by using ordered facility selection words. Usage of other facilities is not equalized and nonordered facility selection words are used.

3.3.4.1 Nonordered facility selection words. The busy/idle state of an individual facility is represented by one bit in a facility selection word of that type. The bit number within the facility selection words corresponds to the hunt number of the facility. When an idle facility is to be selected, a linear search is made of the busy/idle bits until an idle facility is found. For cases where it is not desirable to select the same facility in periods of light traffic, a random starting point is used for the search. This also results in roughly equalizing usage of the facilities.

3.3.4.2 Ordered facility selection words. When use of facilities of a type is to be precisely equalized, ordered selection words are used. Eacb ordered huntable facility is assigned two bits, a busy/idle bit and an *order bit*. The order bits are searched for facilities next in line for selection; the busy/idle bits are used to record which facilities have become idle after selection. Hence, the true availability state of each facility is represented by a combination of the two bits.

3.4 Buffers and hoppers

Base level and IO25 interrupt programs communicate witb eacb other via call store *buffers* and *hoppers*. Generally, buffers are used to pass information from base level programs to interrupt programs, wbcreas hoppers are used to pass information to base level programs.

3.4.1 Peripheral order buffers

A *peripheral order buffer* (POB) is associated with each network time slot. Each POB contains the equipment locations of tbe incoming trunk and tbe announcement track trunk to be connected during the next announcement sequence (Fig. 7).

POBs are loaded by a call processing program at base level and are unloaded at the time the corresponding announcement connections are made at IO25 interrupt level. Constructing announcement connections

ANNOUNCEMENT CONTROL	ANNOUNCEMENT TRACK EQUIPMENT LOCATION
TIME SLOT NUMBER	INCOMING TRUNK EQUIPMENT LOCATION

Fig. 7—Peripheral order buffer (POB) format.

at base level saves IO25 interrupt time. The connections are made at IO25 interrupt level to properly synchronize announcement connections machine phrase timing. All other orders to make connections are issued at base level upon demand because the timing requirement is not as stringent.

3.4.2 Autonomous scan hoppers

During the IO25 interrupt, a program interrogates the master scanner ferrods associated with each autonomous scanner to determine if a trunk supervisory change of state has been reported. If a message is present, the IO25 program loads it into the first vacant entry of the *autonomous scan hopper* (Fig. 8). There is one autonomous scan hopper per network, each containing 10 entries. To conserve IO25 interrupt time, each message stored in this hopper is recovered and processed at base level.

3.4.3 Directed scan hoppers

A *directed scan* is used generally to determine on demand the supervisory state of a trunk. In addition, a directed scan is performed to obtain the type of call intercepted on a three-class operator number identification (ONI) trunk (i.e., whether the call is a regular intercept, blank number, or trouble intercept). This is possible since three-class ONI trunks can inform the AIC via dc signaling of the class of intercept; this dc signal is registered on two master scanner ferrods when the trunk is directly scanned.[3]

Requests for directed scans of trunks are initiated at base level by placing the number of the CIPR associated with the request into a *directed scan hopper*. In a subsequent IO25 interrupt the request is

M	CONTROL INFORMATION	OFF-HOOK BIT	ON-HOOK BIT	EQUIPMENT LOCATION OF TRUNK THAT CHANGED STATE

M = MESSAGE PRESENT FLAG

Fig. 8—Autonomous scan hopper format.

removed and a directed scan order issued to the associated connector and scanner. The answer to this order is retrieved about one millisecond later during the same IO25 interrupt and is stored in the CIPR.

3.5 Input-output registers

Certain input-output functions requiring frequent attention are performed autonomously by control unit circuits. The performing of these functions via direct program actions would consume excessive real time. The information obtained from and used by these circuits are stored in call store *input-output registers.*

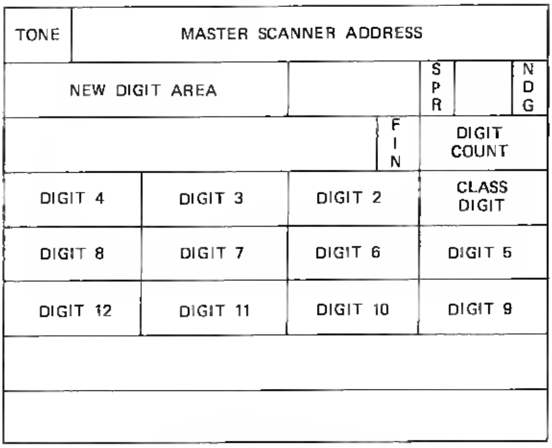
3.5.1 Originating registers

One *Touch-Tone* receiver and several multifrequency receivers are provided in an AIC for digit receiving. The scanning of digit receivers for new digits is an input-output function that is performed at a 10-millisecond rate autonomously by control unit circuits. These circuits monitor the ferrods assigned to the digit receivers and, when digits are received, store the new digits in call store records called *originating registers* (ORs). During alternate IO25 interrupts, the digit receiving program stores the new digits that have been placed into ORs, permitting another new digit to be received in each OR. When a digit receiver is connected, software linkage is established between a CIPR and the OR corresponding to the digit receiver.

One OR is *dedicated* to each digit receiver and it can store all digits received from a local office or an operator position on any intercept call. The first word of each OR contains information which identifies the master scanner row and ferrods associated with the digit receiver permanently assigned to that OR. The format of an OR is shown in Fig. 9.

3.5.2 Data outpulsing buffers

A 64-word call store *data outpulsing buffer* is used for the storage and transmission of up to 16 data outpulsing messages (eight per switching network) for display at CIB operator positions. These messages are loaded by an IO25 interrupt program and transmitted by control unit circuits at a rate of about 800 bits per second. A data message is stored vertically, one bit per word, in storage slots referred to as *data channels*; one data outpulser is permanently associated with each channel. Circuits transmit one word every 1.251 milliseconds, where each word contains one bit for each of the 16 data channels. The transmission



TONE = MF OR TOUCH-TONE® RECEIVER
SPR = SIGNAL PRESENT FLAG
NDG = NEW DIGIT FLAG
FIN = RECEIVING FINISHED

Fig. 9—Originating register (OR) format.

over each data channel is continuous; when no message is present, nulls are transmitted.

Since the circuits are arranged to transmit only 64 bits per message, the transmission of the 94-bit data message to a CIB operator position must be done in two parts. The first part of the message is loaded into the outpulsing buffer slots of a channel. The remaining bits are stored vertically in a supplementary outpulsing buffer. After approximately 40 bits have been transmitted (i.e., two IO25 interrupts later), the rest of the message is constructed by appending the bits in the supplementary outpulsing buffer to the remaining untransmitted bits in the outpulsing buffer.

3.6 Queue structure and control

Most calls bidding for facilities are placed in queues. This allows facilities to be distributed in a predetermined order of priority based on the importance of the call type. Exceptions to this rule are the assignment of CIPRs which is done on a first-come, first-served basis and the assignment of multifrequency (MF) receivers and time slots to new call originations on which digit receiving is expected (Section 4.2.2.1).

Each switching network has a block of 128 call store words reserved for call queuing. This block is divided into 12 queue categories which

N	TIME SLOT NUMBER	P	CIPR NUMBER	H

N = NETWORK NUMBER
P = INDICATION OF TRAFFIC PEG COUNTER INCREMENTED
H = ON-HOOK/OFF-HOOK BIT

Fig. 10—Queue word format.

are used for different classes of calls. Tbe lengtb assigned to the various queues is dependent upon the traffic mix bandled by the AIC. In addition, in the call store queue block of the first network, a 13th queue is used to store disc file lookup requests for calls appearing on incoming trunks appearing on either switching network. Except for the disc queue, queues are assigned an order of priority. Higher-priority queues have access to facilities before lower-priority queues bidding for the same facilities. Within a queue, calls are served on a first-come, first-served basis when facilities become available. Each queue entry is one word and contains several items of information as shown in Fig. 10. Associated witb each queue is a four-word control block shown in Fig. 11.

3.7 Other semipermanent data areas

3.7.1 NNX-ABX translation table

The NNX-ABX translation table gives an AIC the capability of announcing local office codes as two letters and a digit (ABX) instead of as three digits (NNX). Each entry in this table contains the three NNX digits, a code which is translatable into the Numbering Plan Area (NPA) of the local office, and control bits which indicate the circumstances when ABX should be announced. Each entry also contains the equipment locations of the associated alphabetic announcement tracks for the NN digits of the local office code.

3.7.2 Miscellaneous nongeneric data

In addition to tbe previously mentioned data, several other items of nongeneric data are needed to define an AIC. These include codes

QUEUE WRITE ADDRESS
QUEUE READ ADDRESS
QUEUE FIRST LOCATION
QUEUE LAST LOCATION

Fig. 11—Queue control block format.

which identify the AIC and the NPAs served by it, indications of the optional features provided for at the AIC, and indications of the amount of optional equipment at the AIC.

3.8 Nongeneric backup data

Since the AIS stored program is totally generic, each installation has nongeneric data which define the way it is equipped. Under normal operating conditions, the data which are stored in the semipermanent data areas of call store are durable. However, system design errors or hardware faults could lead to circumstances which could destroy the duplicated call store data. Therefore, it is not prudent to depend on call store alone for storing the nongeneric data. In an AIC two electrically alterable disc files are otherwise required to store records of intercepted numbers. Accordingly, a small section of both of these files is used to provide backup for the nongeneric data. In case nongeneric call store data are mutilated, they can be reinitialized quickly from a disc backup (Section VI). This backup data base also provides a convenient and expedient method for making changes to nongeneric data (Section 7.2).

When nongeneric data are placed in call store, considerable redundancy is introduced to increase the efficiency of call processing programs. The format of the file backup version of the data is quite different from that of call store. Rather than providing an image of call store, file information is organized in a rigid, virtually nonredundant format. When reinitializing call store semipermanent data, many disc file accesses are required to obtain the backup data. The time required to perform each access is an order of magnitude greater than the processing time required to reinitialize call store data from the file data obtained during that access. Thus, by eliminating redundancy, the time necessary to restore the system from backup data is minimized. Furthermore, this data organization lowers the probability of inconsistencies within the backup data structure, which could lead to conflicting and inconsistent action by the audit programs (Section V).

Nongeneric data are stored on one track of each disc file using a maximum of twenty-three 20-word blocks, each word containing 42 data bits. The number of blocks required for a given AIC depends on the number of switching networks equipped and the number of NNX-ABX translations. All but two blocks contain trunk or NNX-ABX translation information, so that packing schemes have been developed to most efficiently pack these classes of data. The remaining blocks contain all other nongeneric data.

3.9 Register linkages

Many call store registers, some associated with hardware facilities, are needed to successfully process intercept calls. Many of these registers are closely related and permanently linked to each other, as shown in Fig. 12. In addition as calls are processed other transient relationships develop between the registers. Pointers, redundant for call processing expediency, are introduced into call store to link these registers together. An example of these transient linkages for operator-subscriber connections is also shown in Fig. 12.

IV. CALL PROCESSING PROGRAMS

The call processing programs control the processing of intercept calls routed from local switching offices. This section describes the functions performed by these programs.

4.1 Call processing program organization

4.1.1 Basic call processing functions

Call processing functions are accomplished by a series of highly interrelated programs, each of which performs a specialized task. These tasks include:

(*i*) detection and processing of trunk supervisory changes (Section 4.2);
(*ii*) reception and analysis of sequences of multifrequency digits (Section 4.3);
(*iii*) communication with the disc files to determine the disposition of intercept numbers (Section 4.4);
(*iv*) determining and establishing sequences of announcement connections to incoming intercept trunks and operator positions (Section 4.5);
(*v*) connecting incoming trunks to operator positions and reacting to operator keying actions (Section 4.6);
(*vi*) performing call disconnects (Section 4.7); and
(*vii*) performing functions ancillary to call processing (Section 4.8).

The relationships among these functions are best illustrated by examining a typical call (Fig. 13). Consider an intercept call which is to be routed from a local office equipped with *automatic number identification* (ANI) features, i.e., equipped to automatically identify the called number.* A trunk to the AIC is seized at the local office. When

* This differs from the standard use of ANI, which is to identify the calling number.

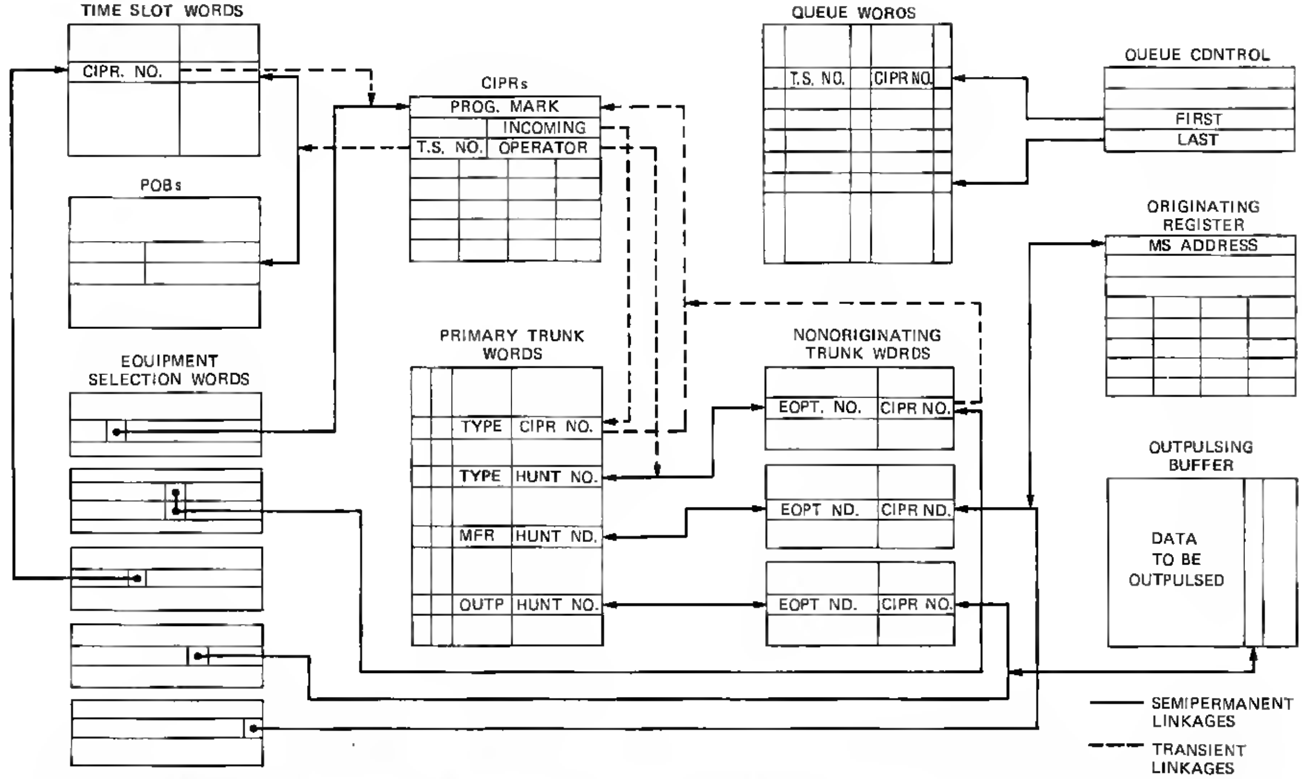


Fig. 12—Semipermanent call store linkage structure and transient linkage structure for connection between operator position and incoming intercept trunk.

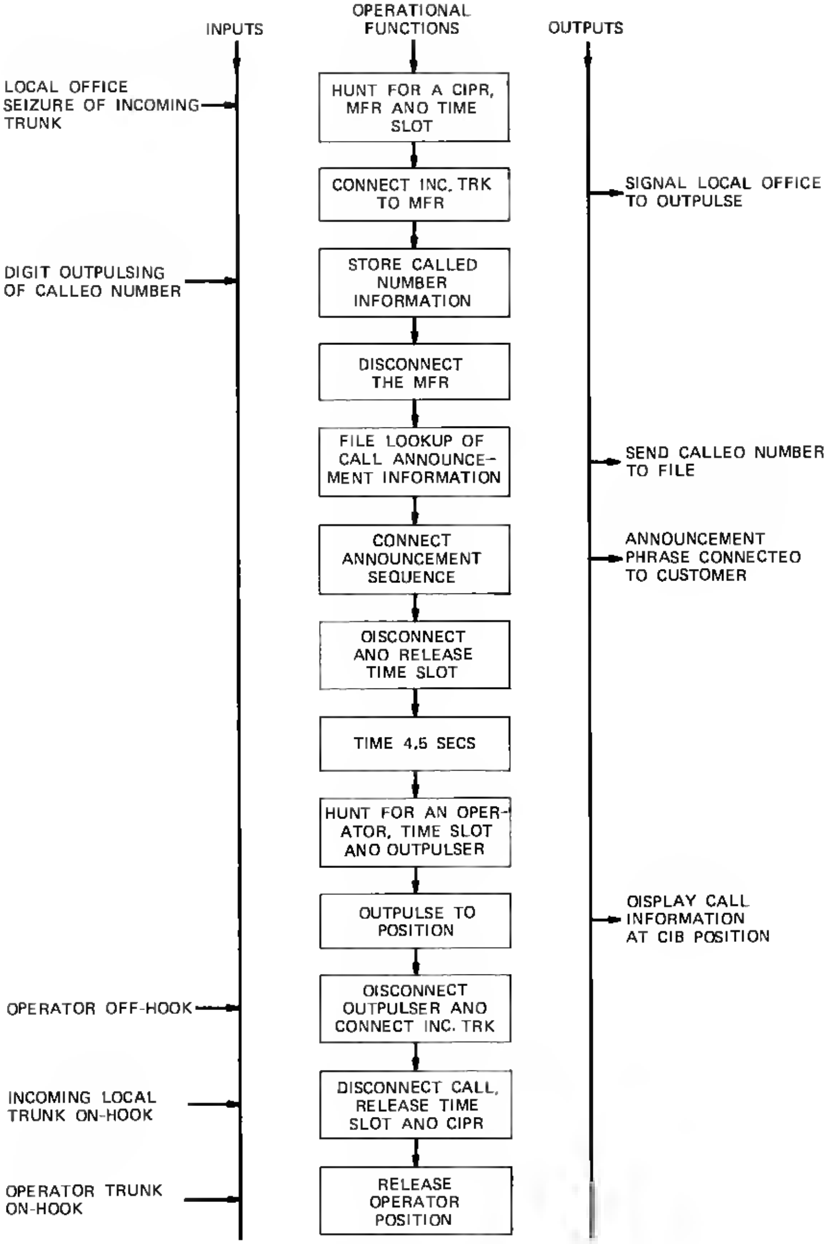


Fig. 13—Typical ANI call sequence.

the seizure is recognized at the AIC, an MF digit receiver is connected in a network time slot to the incoming intercept trunk. This connection causes tbe MF receiver trunk to send a *wink* signal[3] to the local office, indicating that the MF receiver is connected. The local office then outpulses a stream of MF digits, identifying tbe called number and the class of intercept call.

When all digits have been received, the MF receiver is disconnected and the digits are analyzed. If required, a disc file lookup is performed to determine the intercept number disposition. If the lookup indicates that an announcement sequence should he connected, the incoming trunk is connected to a series of 0.5- and 1.5-second announcement pbrases. This informs tbe caller of the intercepted number status and of a new number, if available.

After the announcement, a delay of 4.5 seconds is provided to permit disconnect. If the caller remains on the line, the intercept trunk is connected to a CIB operator position for operator assistance. All details regarding the intercepted number are outpulsed and displayed to the operator. Access to additional information via the disc files is available to the operator upon keying a suitable sequence of MF digits. The call is disconnected when tbe incoming intercept trunk goes on-hook.

4.1.2 Call processing program structure

The structure of the call processing programs is depicted in Fig. 14. These programs are executed at 1O25 interrupt level and at hase level. The 1O25 interrupt programs interface the software with autonomous

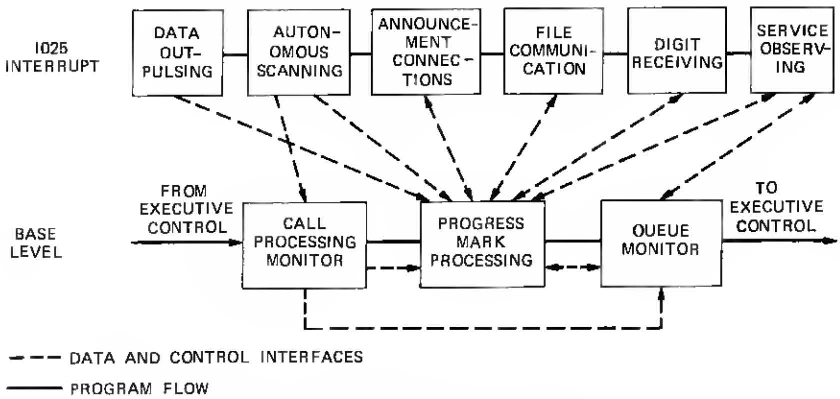


Fig. 14—Call processing program structure.

hardware. In general, base level programs process input signals passed from IO25 interrupt programs and generate outputs to be transmitted by IO25 interrupt programs.

The call processing monitor controls the flow of the three base level call processing programs: trunk supervisory change, progress mark processing (Section 3.2), and queue administration. These programs are sequenced to minimize the elasped time before trunk supervisory changes are processed and to maximize the first-come, first-served use of available facilities. Accordingly, trunk supervisory change processing is performed first; this may involve setting up initial connections, placing calls in queue, or passing inputs via the CIPRs to the progress mark programs. Progress mark routines are executed next, before the queue administration program, so that facilities released by the progress mark routines can be used to serve calls in queue. The queue administration program executes last. It selects available facilities for calls in queue, and then updates the progress marks for further processing actions in the next main program loop. This sequence insures that calls in queue will have access to facilities before new originations.

The IO25 interrupt programs are sequenced so that critical time-dependent functions are executed first. As an example, Section 3.5.2 relates the close interaction between circuits and IO25 interrupt programs to accomplish data outpulsing. This interaction is precisely timed to insure that the second segment of the outpulsed message is properly appended to the first segment.

4.2 Supervisory change detection and processing

Supervisory changes of state (between on-hook and off-hook) are detected on incoming trunks from local offices and from remote AICs, on trunks from and to operator positions, and on trunks to home AICs. Such changes inform the program of call originations and disconnects, and of changes of service state of operator positions and trunks to home AICs. The program reacts to each change of state by performing an appropriate processing action.

4.2.1 Detection of supervisory changes

Changes of supervisory state are reported by a combination of hardware and software actions. The autonomous scanner in the time-division network monitors trunk supervisory states by sequentially comparing the present states of trunks with those recorded in its last look memory.[3] When a change is detected, the autonomous scanner

stops and indicates on master scanner ferrods the equipment location of tbe trunk and the direction of the change.

An IO25 interrupt program interrogates the master scanner once every 25 milliseconds for autonomous scanner messages. If a message is present, the information is placed in tbe autonomous scan hopper for further processing (as described in Section 3.4.2). In this manner, one change of state per switching network can be reported in each 25-millisecond interval. This provides for sufficiently rapid trunk scanning, independent of tbe length of the main program loop.

4.2.2 Processing of supervisory changes

Initial processing of supervisory changes is performed during the IO25 interrupt. In general, the change of state is kept in the autonomous scan bopper for base level processing and the autonomous scanner is restarted with its last look memory updated.[3] However, if tbe cbange is an off-book indicating a new seizure of an originating trunk (sucb as an incoming intercept trunk), some additional processing is performed. A CIPR is selected and call store linkage is established from the primary trunk word to the CIPR. In the event tbat no CIPR is available, the origination cannot be accepted. In this case, the autonomous scanner is restarted but the last look memory is not updated. Tbe origination may be detected again on a later autonomous scan cycle. Delaying the acceptance of new call originations at the AIC provides a natural defense against overloading system processing capabilities.[1]

Further processing of trunk supervisory changes is performed by base level programs. The actions taken by these programs depend on:

(*i*) tbe type of trunk that changed state;
(*ii*) the direction of the supervisory change; and
(*iii*) wbether or not tbe trunk was associated with a call and, if so, the type of call and the stage to which it had progressed.

Base level actions are described in more detail in tbe following sections.

4.2.2.1 Processing changes on incoming trunks. Incoming trunk cbanges from off-hook to on-book are reported by the base level trunk supervisory program in tbe CIPR associated with the trunk. This action results in the progress mark program disconnecting the call and releasing the CIPR for anotber call (Section 4.7).

Tbe action taken on on-hook to off-hook cbanges of state (call originations) depends on the trunk type. In general, a time slot is as-

signed to tbe call and a connection is made between the incoming trunk and nonoriginating equipment (e.g., digit receiver or outgoing trunk to an operator position). Tahle III contains details of actions taken on call originations, including a numher of exceptions to the above.

4.2.2.2 Processing changes on outgoing trunks to operator positions. The actions taken on changes of state of outgoing trunks to operator positions depend on whether or not a CIPR has heen assigned. If the trunk state changes from on-hook to off-hook witb no CIPR assigned, this indicates the operator has placed the position in a *made busy* state and the appropriate bit in a facility selection word is set indicating this position is temporarily not availahle.

Changes of state on trunks that have a CIPR assigned are reported in the associated CIPR for action by the progress mark programs. When an off-hook to on-hook change of state is reported, the position is eventually idled (Section 4.7). Conversely, an on-hook to an off-hook change of state usually verifies that the operator is talking on a connection (Section 4.6).

4.2.2.3 Processing changes on outgoing trunks to home AIC. Calls requiring CIB operator assistance at remote AICs are routed via outgoing trunks to the home AIC. When a connection is made at the home AIC hetween an incoming trunk from a distant AIC and a CIB position, a signal is passed which causes the outgoing trunk at the remote AIC to go off-book. The call processing program at the remote AIC then outpulses the information necessary for the CIB display. After a timed interval, the incoming intercept trunk is connected to the CIB operator position. After the call has heen completed, disconnect actions are similar to those performed at a home AIC.

4.3 Digit receiving and analysis

Digit receiving programs accumulate, decode, and analyze sequences of MF and *Touch-Tone* digits. Analysis of a sequence of digits is complex, since the digits can he received from various sources and be associated with various stages of calls. Sequences are received at the AIC from:

(*i*) ANI-equipped local offices, when MF outpulsing the intercept numher;

(*ii*) CIB and ONI operator positions, when MF keying the called number on calls intercepted in non-ANI local offices (Sections 4.6.1 and 4.6.3.2);

Table III — Actions taken on call originations

Type of Trunk	Initial Actions		
	All Facilities Available	Time Slot Available; Service Circuit or Operator Position Not Available	Time Slot Not Available
Incoming Intercept (ANI)	Connect to MF receiver (Section 4.3)	Queue for time slot and MF receiver	Queue for time slot and MF receiver
Incoming Intercept (1-Class ONI)	Connect to ONI operator position (Section 4.6.1)	Queue for ONI operator position; apply audible ring	Queue for time slot and operator position
Incoming Intercept (3-Class ONI)	Regular intercept—connect to ONI operator position	Queue for time slot and ONI operator position	Queue for time slot and ONI operator position
Determine Call Class by Directed Scan (Section 4.6.1)	Trouble intercept—connect to trouble operator position	Queue for time slot and trouble operator position	Queue for time slot and trouble operator position
	Machine intercept—connect to blank number announcement	Not applicable	Queue for time slot
Incoming from Remote AIC	Queue for CIB operator position (Section 4.6.3)	Queue for time slot and CIB operator position	Queue for time slot and CIB operator position
Incoming from CIB Operator	Connect to MF receiver (Section 4.3)	Queue for time slot and MF receiver	Queue for time slot and MF receiver
Incoming Operator Training	Connect to *Touch-Tone* receiver (Section 4.8.3)	Wait for *Touch-Tone* receiver and release time slot	Wait for *Touch-Tone* receiver and release time slot
Incoming Directory Assistance	Queue for time slot and CIB operator position (Section 4.6.3.2)	Not applicable	Not applicable

(*iii*) CIB operator positions, when keying MF digits requesting additional displays or announcements of intercept number information (Section 4.6.3.1.) or when connecting a service assistant on the call (Section 4.6.4);
(*iv*) a trainer when entering *Touch-Tone* digits while training a CIB operator (Section 4.8.3);
(*v*) CIB operator positions, when keying MF digits upon vacating or reoccupying a position (Section 4.6.3); and
(*vi*) ANI/ONI concentrators, when identifying the type of ONI call (Section 4.6.1).

In response to certain types of trunk supervisory state changes, an appropriate digit receiver is connected in a time slot, causing a signal to be returned by the digit receiver trunk to the transmitting source. This indicates that a receiver is connected and outpulsing can begin.

4.3.1 Digit reception

Each MF digit transmitted to the AIC is indicated in a 2-out-of-6 code on master scanner ferrods and subsequently placed in an OR as described in Section 3.5.1. *Touch-Tone* digits are handled similarly except that they are transmitted in a 1-out-of-4, 1-out-of-4 code. Checks are performed at IO25 interrupt level on the digit sequence to determine whether or not the sequence is valid.

While the IO25 interrupt digit receiving program is storming digits in an OR, the base level digit analysis program is checking for the completion of digit receiving. When the start pulse (ST) is received indicating the end of the digit stream, the digit receiving program notifies the analysis program that all digits have heen received. The digits are then transferred to the associated CIPR. If the digit outpulsing is not completed within 24 seconds or if the IO25 interrupt digit receiving program has indicated an error in receiving, the call is placed in queue for a CIB operator. When the call is connected to the operator position, an indication, AIC FAILURE, is displayed at the position. The error is also reported to maintenance programs so that diagnostic tests may be performed on the digit receiver. If no error has occurred the digits are analyzed.

4.3.2 Digit analysis

The interpretation of a digit message is done at base level by progress mark digit analysis programs. Analysis depends on the source of the digits, the class digit contained at the beginning of the digit stream and

Table IV — MF and *Touch-Tone* keying sequences

Source	MF Sequences	Explanation and Disposition
Local office via ANI trunk	(KP) 0 (ST)	Machine intercept; give blank number announcement.
	(KP) 1 (ST)	Trouble intercept; connect to trouble operator position for assistance.
	(KP) 2 (ST)	ANI failure; connect to CIB operator position for assistance.
	(KP) 3 NXX XXXX (ST)	Regular intercept; perform disc file lookup to determine called number disposition.
Local office ONI calls via ANI/ONI concentrator	(KP) 5 (ST)	Announcement previously given at ANI/ONI concentrator; route to CIB operator for post-announcement assistance.
	(KP) 6 (ST)	Regular ONI intercept; connect to ONI operator position for number identification.
	(KP) 7 (ST)	Machine intercept; give blank number announcement.
	(KP) 8 (ST)	Trouble intercept; connect to trouble operator position for assistance.
CIB operator position	3 (ST)	To SA key depressed; establish connection to service assistant console.
	3 01X (ST) 3 01XNXX (ST)	Operator-keyed trouble report; generate printout at maintenance teletypewriter.
	3 1YX (ST)	Test call.
	3 NXX XXXX (ST) 3 NYX NXX XXXX (ST)	If KP ONI key depressed, regular intercept call identified by operator; if KP ANN key depressed, CIB operator requesting additional announcement; in either case, perform disc file lookup and give announcement.
	4 NXX XXXX (ST) 4 NYX NXX XXXX (ST)	KP DISP key depressed; display additional information to CIB operator.
	9 (ST)	Operator has vacated position by removing head set.

Table IV (continued)

Source	MF Sequences	Explanation and Disposition
ONI operator position	3 01X (ST) 3 01XNXX (ST)	Operator-keyed trouble report; generate printout at maintenance teletypewriter.
	3 NXX XXXX (ST) 3 NYX NXX XXXX (ST)	Regular intercept call identified by operator.
	Touch-Tone Sequences	
Trainer while training a CIB operator	(KP) 0 (ST) (KP) 0 NNX XXXX (ST)	Machine intercept; give blank number display at training position.
	(KP) 1 (ST) (KP) 1 NXX XXXX (ST)	Trouble intercept; connect to training position with trouble intercept display.
	(KP) 2 (ST)	ANI failure; connect training position with ANI failure display.
	(KP) 3 NXX XXXX (ST)	Regular intercept; connect training position with post-announcement display.
	(KP) 5 (ST)	Directory assistance call; connect to training position with directory assistance.
	(KP) 6 (ST)	AIC failure; connect to training position with AIC failure display.
	(KP) 7 (ST)	Post-announcement assistance on ONI-originated call; connect to training position with proper display.
	(KP) 8 (ST)	ONI intercept call; connect to training position with display signifying ONI function to be performed.

Key: N = 2, 3, 4, 5, 6, 7, 8, 9
X = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Y = 0, 1

possibly on the first two or three digits of the message itself. Table IV lists the possible keying sequences from each source and gives their interpretation.

Once analysis is completed and the proper call action ascertained, the progress mark is updated. For example, after the MF digit sequence (KP) 3 NXX XXXX (ST) is transmitted from a local ANI office and tbe IO25 interrupt digit reception program bas accepted tbe digits, the base level analysis program begins interpreting the digit stream. Because the digit source is a local ANI trunk, tbe class digit alone determines the action to be taken. In this case, tbe class digit 3 indicates a regular intercept call and the progress mark of tbe call is changed so that a file lookup is initiated. This is described in Section 4.4.

4.4 Call processing disc file communication

4.4.1 Fila access

Regular intercept calls require a disc file lookup based on the seven-digit called number to determine tbe status and any new number information associated with the call. All calls requiring a file lookup are placed in tbe call store disc queue (Section 3.6) containing calls waiting for a file lookup. Tbese requests are entered in this queue by the progress mark routines during base level call processing and are removed at IO25 interrupt level. After it is determined that a file is available for a lookup, the IO25 call processing file communication program removes an entry from the disc queue and sends the called number to the file where the lookup is performed associatively by addressing the file with this number.[4]

4.4.2 Data retrieval and analysis

The average file response time for a lookup is 160 milliseconds.[5] After initiating a file lookup, a program reads a status register in the file control circuit during every subsequent IO25 interrupt to determine if tbe lookup is completed. If tbe lookup is successful, the file reply is contained in the input-output register in the file control circuit and tbe file is made available for otber use. Tbe reply is placed in the associated CIPR; this information normally consists of:

(*i*) a two-digit *status* indicating tbe disposition of the intercept number (for example, the called number is disconnected);
(*ii*) a new number associated with the intercept number, if appropriate; and
(*iii*) new NPA or new locality information, where appropriate.

A file lookup may fail for a variety of reasons. For example, the intercept number may not appear on the file, in which case the caller is connected directly to an operator position with an indicator, NOT IN FILE.

When both files are unavailable due to maintenance conditions, callers are given a blank number announcement (Section 4.5). This announcement is used to provide limited service when the status and new number information are unavailable.

4.5 Announcement connections

The announcement sequence connected on a call depends on the call class and, where appropriate, the results of a disc file lookup. Three types of announcement sequences can be connected:

(*i*) The *blank number announcement* informs the caller that the number he reached is not a working number. It is applied on machine intercept calls, and on regular intercept calls requiring a file lookup when the lookup cannot be performed because neither disc file is available for service.

(*ii*) The *working number announcement* informs the caller that the number he intended to dial is a working number. It is applied on ONI calls when a file lookup results in a not-in-file, a condition most likely caused by a dialing error.

(*iii*) The *"customized" intercept announcement* informs the caller of the disposition of an intercept number as defined in the disc file record. The type of announcement is based on the intercept number status; details of the announcement include the called number and, possibly, a new number. These announcements are applied to regular intercept calls, both ANI and ONI.

Announcements are constructed from 0.5-second digits and 1.5-second phrases. About 50 milliseconds of "quiet" time is provided between sequential announcements on a track and is synchronized between tracks. Announcement connection track switching is performed within this period to prevent "clipping" of announcement phrases. Reference 3 contains a list of the phrases and digits recorded on the announcement tracks.

Timing of announcement tracks is indicated on master scanner ferrods. At the end of each 0.5- and 1.5-second announcement phrase (i.e., during the quiet period), an IO25 interrupt program detects the indication and sends connection orders to the switching network. The identities of the trunks to be connected were previously calculated by

a base level program and stored in POBs (Section 3.4.1). When the IO25 interrupt program completes sending tbe connection orders, it indicates to the base level program that the next POB orders can be constructed. Base level progress mark routines perform this by using tbe intercept number status to reference tbe appropriate *connection table*. Each table entry defines the actions to be taken after each 0.5-second period. The table is indexed by tbe announcement phrase count stored in the time slot word (Section 3.3.1). Tbe information obtained from the table indicates one of the following:

(*i*) no action (e.g., in the middle of a 1.5-second phrase);
(*ii*) the identity of the new announcement track;
(*iii*) a specific reference to called number, new number, NPA, or geographic location information which is required to determine the next track; or
(*iv*) completion of the announcement sequence.

4.6 Operator position connections and actions

Operator positions are provided in AIS installations to assist callers by performing functions not accomplisbed by automatic equipment. Four types of positions may be provided:

(*i*) *ONI operator positions*, used to manually key called number information;
(*ii*) *trouble operator positions*, used to serve trouble intercept calls;
(*iii*) *CIB operator positions*, used to perform a variety of functions such as assisting callers not satisfied with the recorded announcement; and
(*iv*) *service assistant consoles*, used to handle problems which cannot be resolved by CIB operators.

These positions are provided at home and remote AICs as shown in Table V.

4.6.1 ONI operator functions

In an AIS installation witb a large volume of ONI traffic, ONI operator positions may be furnished to receive calls from local offices not equipped with ANI. The ONI operator asks tbe caller the number he dialed and then keys tbe called number into tbe AIC via an MF keyset. Tbe MF receiver used is permanently associated with the operator position; hence, it is *not* necessary to select an MF receiver

Table V — Types of operator positions provided with AICs

Position	Home AIC	Remote AIC
CIB	Must be provided	Never provided
ONI	Optionally provided	Optionally provided
Trouble	Optionally provided	Optionally provided
Service Assistant	Must be provided	Never provided

and connect the ONI position to it through the switching network. Digit receiving and analysis programs store the keyed digits. When the ST digit is received, the ONI operator position is disconnected and thereafter the call is handled like an ANI call.

ONI calls are received from local offices on three types of trunks:

(*i*) one-class ONI trunks;
(*ii*) three-class ONI trunks; and
(*iii*) trunks from ANI/ONI concentrators.

All calls on one-class ONI trunks are routed directly to ONI operator positions. On calls received on three-class ONI trunks, a directed scan (Section 3.4.3) is performed to determine the call class. If a regular ONI call is indicated, the call is routed to an ONI operator position. Otherwise, the call is handled like an ANI call of the same class.

Calls received through ANI/ONI concentrators are first connected to MF receivers. For ONI calls, the concentrator outpulses a digit sequence identifying the call class. Regular ONI calls thus identified are routed to ONI operator positions. Other calls are handled like ANI calls.

4.6.2 Trouble operator functions

Calls to important customer lines, such as hospitals or firehouses, may be intercepted and routed to trouble operator positions if the line has been "plugged up" because of a trouble. The operator consults a printed bulletin and notifies the caller of the disposition of the line. In most cases, the trouble operator offers an alternative method of reaching the desired customer. When trouble operator positions are not furnished or are unoccupied, this function can be performed at CIB operator positions.

4.6.3 CIB operator functions

CIB operator positions are provided only at home AICs. They serve the following functions:

(*i*) further assisting callers on calls receiving recorded announcements or requiring immediate connections to operator positions;
(*ii*) keying the called number and providing a verhal response for ANI-failure calls;
(*iii*) handling directory assistance calls; and
(*iv*) performing ONI and/or trouble operator functions when such positions are not furnished at an AIC.

Calls at a remote AIC may also require CIB operator assistance. Such calls are routed to a CIB operator at a home AIC, using the home AIC as an intermediate switching point. Note, however, that ONI calls to a remote AIC must he served by ONI operators at that AIC.

When a call is to he connected to a CIB position, a data outpulser is connected to the outgoing trunk to the position and a data message is transmitted (as described in Section 3.5.2). This results in a console display containing all availahle details of the call. When transmission is complete, the outgoing trunk goes off-hook, signaling call processing programs to disconnect the outpulser and connect the associated incoming trunk.

A CIB position is equipped with trunk access to each AIC it serves (one home AIC and up to four remote AICs). Using these trunks, the CIB operator may ohtain further information about an intercept call. To do this, the operator depresses a key, causing the trunk to the destination AIC to go off-hook. The call processing programs at this AIC recognize the change of state and connect the incoming trunk from the CIB operator position to an MF receiver. The operator then keys a sequence of MF digits appropriate to the request.

In addition, some operator actions cause MF digits to be outpulsed automatically by circuits within the position. For example, when a CIB operator vacates a position, removal of the headset causes a digit sequence to be outpulsed indicating that the position is now unoccupied.

4.6.3.1 Post-announcement assistance. Callers not satisfied with the recorded announcements may wish assistance from an operator. In such cases, the incoming intercept trunk is connected to a CIB operator position 4.5 seconds after completion of the announcement. The

console display indicates to the CIB operator the call class (regular intercept or operator-keyed), the called number and NPA, the call status, the identity of the originating AIC, and the new number and NPA, if available. While assisting the caller, the CIB operator may desire an announcement or another display of information concerning this (or another) intercepted number. Tbe operator seizes a trunk to an AIC and keys the digit sequence (shown in Table IV). The class digit indicates whether an announcement or display is desired. A file lookup is performed and an appropriate announcement or display results. The display is handled similarly to the initial display. The announcement, which is heard by both the caller and the operator, is connected to the incoming trunk from the CIB operator position. The announcement is disconnected when it is complete or when the operator trunk goes on-hook.

4.6.3.2 Other intercept assistance. Certain types of calls require immediate connection to a CIB operator position. For example, when ONI positions are not furnished or are all unoccupied, CIB positions may receive ONI calls. This function is quite similar to that performed by ONI operators (Section 4.6.1). When an ONI call has been connected to a CIB position, the console display indicates the call class (ONI) and the identity of the originating AIC. Unlike ONI operation, incoming trunks from CIB positions require connection to a hunted MF receiver in a second time slot. This second connection is controlled from a second CIPR. When operator digit keying is complete, the digits are transferred from the originating register to the CIPR associated with the incoming intercept trunk. The facilities used for the second connection are then released.

4.6.4 Transfer of call to service assistant

If the caller requires assistance beyond that provided by the CIB operator, the call may be connected to a *service assistant* (SA). To do this, the CIB operator depresses the TO SA key. A second connection is then established between the incoming operator trunk and the SA trunk. The CIB operator may release from the call by operating the POSITION RELEASE key. This causes a transfer of call data from the CIPR associated with the original connection; a new connection is established between the incoming intercept trunk and the outgoing trunk to the SA console. The CIB position is made available for another call.

4.7 Call disconnect

Certain types of trunk supervisory changes (Section 4.2) stimulate call disconnects. In general, when a trunk goes on-hook, the associated call is disconnected. The additional actions taken in response to an on-hook depend on the type of trunk that changed state, and the type and supervisory state of the trunk to which it is connected, if any.

When a connection has been established between an incoming trunk (from a local office or a remote AIC) and an outgoing trunk to an operator position, an on-hook report on either trunk causes the call to be disconnected. If the incoming trunk goes on-hook first, all associated facilities, hardware and software, are idled except for the operator position. The outgoing trunk to the operator position is placed in a made-busy state until it goes on-hook. Normally, this on-hook occurs momentarily; however, should the position have been manually made busy, the trunk will not go on-hook until the position is again ready to receive calls.

The operator may release a call from the position manually by operating the POSITION RELEASE key. This causes the outgoing trunk to the operator position to go on-hook first. In this event all facilities are released except the CIPR and the incoming trunk. The CIPR is held until the incoming trunk goes on-hook or until two minutes have elapsed. After two minutes, a permanent signal on the incoming trunk is suspected, and maintenance programs are notified to perform testing of the trunk.[3]

On all connections other than caller to operator, disconnect actions are based only on the state of the incoming trunk. For example, when an incoming trunk connected to an MF receiver goes on-hook, the supervisory state of the MF receiver trunk is ignored. The call is simply disconnected and all associated facilities are idled immediately. Similarly, if an incoming trunk goes on-hook when it is not connected in a time slot, associated facilities are released and the call is removed from any queue in which it may be entered.

4.8 Ancillary call processing functions

Several functions are performed by call processing programs which are not central to the job of processing intercept calls. These functions are incorporated into system design in a way that does not interfere with expedient handling of intercept traffic.

4.8.1 Service observing

Service observing provides a means for monitoring the performance of an AIS, in general, and of CIB operators, in particular, to determine

the quality of service offered. This allows telephone companies to ensure that intercept service standards are being achieved.

One call (at a time) to a CIB or ONI operator position may be monitored at a remote *service observing* (SO) *desk*. The observer receives via a data link three types of information concerning the monitored call. These are:

(*i*) all data which are displayed at the operator position; these are printed on a paper tape at the SO desk;
(*ii*) reports of all actions taken by the operator sucb as MF keying; these are shown on lamps and paper tape at tbe SO desk; and
(*iii*) the conversation which transpires between the caller and an operator and between the caller and a service assistant, if any.

When a period of monitoring calls is to begin, the service observer indicates this to the AIC by key operation. In response, when monitoring is active and the SO desk is idle, the progress mark program selects a call to be monitored from a CIB operator queue. At this time a *position called* signal and all data to be displayed at the CIB position are sent to the SO desk. After the incoming trunk is connected to the operator position, a *position attached* signal is sent to the SO desk. A special, three-way connection[3] is also established to allow the observer to monitor the conversation. All additional actions taken on the call are transmitted to the observer until either the call is disconnected or the observer releases the position. Another call can then be sent to the SO desk for monitoring.

4.8.2 Operator-keyed trouble reports

Problems encountered by CIB and ONI operators while attempting to assist callers may point to trouble conditions in the AIS or connecting offices. If tbese conditions are reported quickly and accurately to maintenance personnel, the reports can be used in conjunction with other indications to resolve the problems. For this reason, these operators are able to key reports of ten different types of trouble conditions into the AIC. The format of the digit sequence and a list of the trouble conditions are given in Table IV and Table VI, respectively.

After the reception and analysis of the digits, a base level progress mark program generates a printout on a maintenance teletypewriter. This printout contains the information keyed by the operator, the type and number of the operator position, and all pertinent information about the connection involved.

Table VI — Trouble conditions reported by operators

Keyed Code	Description
010	Caller reports or operator detects difficulty in hearing or being heard.
011	Operator detects abnormal noise on connection.
012	Caller reports or operator detects an announcement phrase that is not clear.
013	Partial or mutilated display.
014	Position seizure with valid display. No caller response to operator challenge and no room noise heard.
015	Caller on line with display from previous call, and no zip tone.
016	More than one caller on line.
017	Operator hears another operator on position.
018	For no apparent reason, the connection between the caller and the operator is broken during conversation.
019	Code assigned for local use in analyzing special trouble conditions.

4.8.3 Operator training

Facilities are provided at the AIS for training CIB operators to perform all appropriate functions. Up to two positions at a CIB may be equipped on a plug-in basis with standard 12-key *Touch-Tone* card dialers. Activation of training is indicated to the program by a start-training message which is entered via the card dialer.

On each training call, the trainer enters via the card dialer a number used only for operator training. The digit sequence is received and analyzed and the call is routed directly to the trainee with the proper display. All call conditions normally encountered by a CIB operator may be simulated in this manner. On a call where an announcement is normally required, a file lookup is performed but no announcement is constructed; the post-announcement display is simply sent to the position. Training continues until a stop-training message is received.

4.8.4 System status indicators

A *lamp signal cabinet* is provided at both the CIB and ONI position locations to indicate to the operators traffic and trouble conditions in the AIS.[3] For example, whenever calls are waiting for operators, a lamp is lighted indicating the condition.

These lamps are lighted and extinguished by the call processing

programs hy means of data outpulsing. An outpulser is connected to a data trunk outgoing to the lamp signal cabinet. A code is loaded into the data outpulsing area and outpulsed as described in Section 3.5.2. Decoding circuits at the cabinet then light and extinguish the lamps indicated.

V. CALL STORE AUDITS

5.1 Audit philosophy

The AIS program depends heavily on data in its call store memory to record the states of calls and of system hardware and software resources. Hardware errors, program bugs, and incorrect manual operations can mutilate data in call store, causing calls to be mishandled and leaving system resources in unusable states. In addition, data errors could propagate throughout the call store data causing service to degenerate, possibly creating the need for a system initialization (Section VI) to recover from the errors.

Some of these errors are eliminated hy defensive programming techniques. However, some types of errors would require a prohihitive amount of processor time to prevent and still other more subtle errors could not be readily found using defensive programming techniques. Hence, *audit programs* are needed to protect the AIS from the effects of data mutilation. These programs detect and correct errors in the critical transient and semipermanent call store memory areas such as call-in-progress registers, queues, records of connections in the network, and trunk translation data.

The audit programs employ the philosophy of using the external hardware states and nongeneric data from the files as references. This eliminates the need to make any assumptions about the correctness of call store data hefore auditing and simplifies corrective techniques since normally many more errors occur in call store than in the external data. For example, the audit of the basic call processing linkage structure starts by reading the connector and scanner time slot memory to determine the equipment locations of the trunks actually connected. Call processing software records are then compared with the time slot readout information for consistency. If errors are found, audits report them via teletypewriter messages and correct them hy returning the call store memory and hardware to consistent, viahle states. It should be noted that certain inconsistencies reflect errors in the external information and the audits can detect and report these. A high rate of such errors can trigger maintenance action on the associated peripheral circuit.[3]

Other audit programs depend on the nongeneric data stored on disc file or on trunk supervisory state scanning for external information. Since various audits use different reference points and limit the areas of memory which they audit, the combined action of several audits, some triggered by others, may be necessary to clear a data problem and halt error propagation.

5.2 Types of audits

Audits may be divided into three groups: *transient data audits*, *semipermanent data audits*, and *timeout audits*.

5.2.1 Transient data audits

Transient data audits generally detect and correct linkage errors in call store registers associated with calls in the system. This is done by comparing redundant information. Redundant information is that which is represented in different forms within the call store memory for call processing expediency or indicated by the states of peripheral circuits; this is distinct from semipermanent information for which a backup copy exists. A brief description of a transient data audit follows.

The *time slot memory audit* checks the correctness of the basic call processing software linkages (Section 3.9). A word is read from the connector and scanner time slot memory. The equipment locations of the connected trunks are also read from the call-in-progress register indicated by the time slot word. Then, the facility control words corresponding to these trunks are read. If the linkages, as shown in Fig. 15, are not consistent, the call is disconnected and the associated trunks and CIPR are released. In this way, the call records are checked using the redundancy inherent in the call store linkage structure and

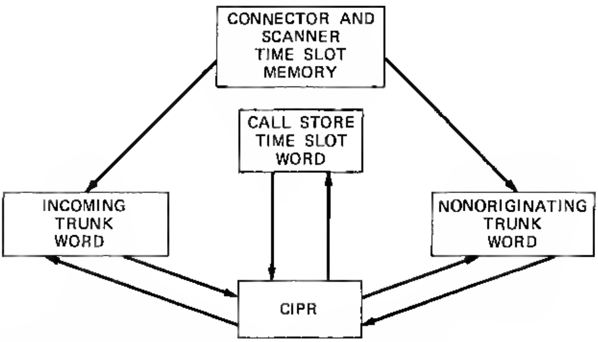


Fig. 15—Abstract representation of linkages audited by time slot memory audit.

the peripheral circuits. In addition, the audit also checks the status of the time slot facility selection words with the contents of the time slot memory; any inconsistencies found are corrected.

Three other transient areas of call store are audited. The *nonoriginating equipment audit* compares the facility selection words for all nonoriginating equipment with the actual state of the associated equipment to prevent equipment from becoming lost to the system or multiply used. The *queue audit* checks the linkage structure of calls in queues to prevent these calls from being mishandled. Finally, the *connector and scanner maintenance words audit* checks call store words associated with the state of the connector and scanner circuits with the actual states of the circuits.

5.2.2 Semipermanent data audits

Semipermanent data audits detect and correct errors in the areas of call store where nongeneric data particular to an AIC are stored. This is done by comparing call store records with a disc file backup record. For example, call store records of NNX-ABX translations are periodically compared with a disc backup record. When a disagreement is found, the file record is assumed correct and call store is corrected.

When nongeneric data are altered via plant changes, the same audits are used to regenerate semipermanent call store data without disrupting service. This is accomplished hy overwriting only that information which has been changed on the disc file backup record (Section 7.2). In addition, the semipermanent data audits are used to place nongeneric data in the call store memory during phase B and C system initializations (Section 5.4). The use of the semipermanent data audits and their relation to other operational programs is shown in Fig. 16.

Another audit compares the nongeneric data on the two disc files and makes some internal consistency checks on the data. If a discrepancy occurs, the error is reported and manual correction is required; semipermanent data audits are suspended until the errors are corrected. This audit enhances the integrity of the disc file backup records of nongeneric data.

5.2.3 Timeout audits

Timeout audits monitor continuous use (without release) of a software or hardware facility. If a predetermined time period is exceeded, a timeout audit either verifies the correct use of the facility or releases it. For example, a check is made to determine if a CIPR is active for a period of time longer than the maximum holding time for the type

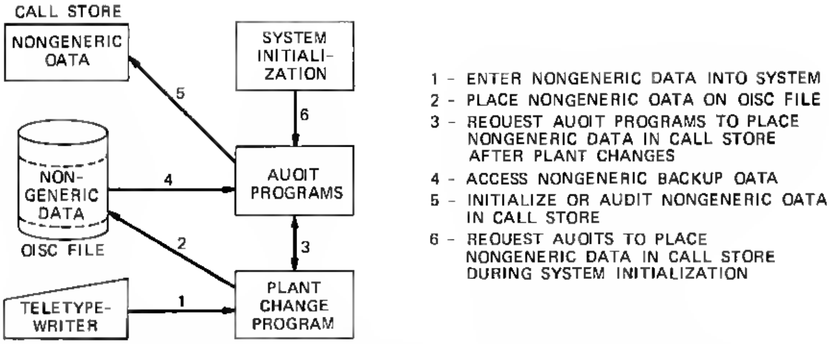


Fig. 16—Interrelationship between audit, plant change, and system initialization programs.

of associated call. If the time is exceeded, the system may have lost control over that CIPR and action is taken to verify the CIPR usage or correct the situation.

5.3 Audit control

Audits are executed once every main program loop. Upon gaining control from the executive control program, the audit monitor causes any requested client audit to be executed based on a predetermined order of priority. The progress of a client audit is then controlled through a segmenting structure until it has completed.

Individual client audits are requested by the monitor on a routine, periodic basis. The rate at which different audits are run was chosen to maximize the call processing, administrative, and maintenance needs of the system while minimizing the risk of data mutilation and error propagation. In particular, transient data audits are executed at least once every 40 seconds. At the chosen rates, routine audit requests do not overlap, and an audit segment is executed during about half of the main level loops.

Semipermanent data audits are routinely requested at a rate lower than transient data audits. This is done to prevent excessive file access requests which could delay call processing and hinder use of the disc files by file administration routines. To compensate for this less frequent routine requesting, the transient data audits typically request the semipermanent data audits when suspect semipermanent data are encountered. Other system programs also request audit programs if suspect call store data are encountered.

Most audit programs can be initiated manually via teletypewriter request. Options are available for controlling the detail of error printout, the extent of the audit, and whether the audit is to be executed once or repetitively. In addition, messages exist to inhibit and restore all detailed error message printouts and the execution of all audits or a particular client audit.

Executing audits as a major main program loop function rather than as a maintenance program allows the audits to run as asynchronously as possible with respect to all other system programs. This enhances the possibility of detecting all errors, regardless of their sources, and before they propagate.

5.4 Audits during system initialization

The audit programs are used during phase B and C system initializations (Section VI) to place the nongeneric data in call store memory. This is accomplished by running the semipermanent data audits consecutively until the semipermanent portions of call store are completely reinitialized from a disc file backup record. This consecutive running of the audits is called *stitching* the audits together. The stitched mode execution of the audits is requested by the initialization program. During this period, no other system programs are running except those needed to perform file block reads. After all the requested audits have been run, control is returned to the initialization program.

VI. SYSTEM INITIALIZATION

In a complex program-controlled system such as AIS, hardware or software malfunctions can occur which result in improper call processing actions. While circuit redundancy and software corrective techniques such as audits are applied to minimize the effects of such actions, an occasional problem arises which is so serious that severe recovery action, known as *initialization,* is necessary. In addition, similar actions are used for initial startup of the system.

When initialization occurs, all other system activity is suspended for the duration of the initialization. The severity of an initialization determines the degree to which intercept service is disrupted. Three levels of severity, or *phases,* are provided so that increasingly drastic initialization actions can be performed until proper operation is resumed. This structure minimizes over-reaction to troubles which can be cleared without seriously disturbing calls in the system.

In general, the initialization strategy provides a working configura-

tion of a control unit and peripheral units, and brings the call store memory into agreement with the states of system equipment. A control unit switch accompanies all hut manually requested initializations so that a potentially faulty control unit can he switched offline. This control unit then performs all initialization actions. The action taken to initialize the call store memory depends on the phase being executed. Transient, stable, and semipermanent areas of call store can be affected (Section 3.1).

6.1 Causes of initialization

Initialization actions can result from hoth automatic and manual sources. Automatic requests for initialization can be triggered by failure of either hardware or software checks. Among these are:

(*i*) A circuit which times the length of the main program loop and causes an initialization if a maximum time is exceeded. This provides a defense against infinite program looping.
(*ii*) A program which checks that the main program monitors are executed in the correct sequence and causes an initialization if improper sequencing is detected.
(*iii*) A program which frequently tests the online control unit to see that its circuits are operating properly and causes an initialization if a test fails while the control units are not operating in synchronism.

Thus, in general, an initialization results from program insanity or from failure of the online control unit when the control units are not operating in synchronism. Troubles in peripheral units do not normally result in automatic initialization.

Manual requests for initialization are made via the emergency action panel at the maintenance center. Three keys must be operated in sequence so that inadvertent operation of a single key does not result in initialization. In addition, two keys are provided to permit the following options:

(*i*) immediate execution of the most severe initialization phase and zeroing of long-term call store (stable) records not affected by automatic initializations, and
(*ii*) placing the system in a noncall processing mode during which file backup records of nongeneric data can be initialized or restored from a punched paper tape record (Section 7.2).

6.2 Phase structure

The initialization phase structure and sequencing strategy represent a compromise between maximizing speed of recovery and minimizing disruption of intercept service. Three phases are provided: A, B, and C. They are designed so that most troubles are cleared by a phase A initialization. If phase A does not recover the system, the more drastic phase B initialization may be triggered. Similarly, if phase B is unsuccessful, a pbase C initialization may be executed. If an initialization is triggered within about 20 seconds after a previous initialization is completed, the initialization phase may be escalated.

When a rapid sequence of initializations occurs, phase A is executed once, pbase B is executed twice, then phase C is executed twice. The sequence ABBCC is then repeated until recovery is successful. During each initialization, a different configuration of the control unit with each peripheral unit is establisbed so that a faulty communication path does not prevent a successful initialization. For example, data on the disc files are read during phases B and C to initialize semipermanent data in call store. On four consecutive initializations (two phase B, then two phase C), if both disc files are accessible, all four control unit/disc file access paths are used.

6.2.1 Phase A initialization

The lowest level initialization is phase A. This phase lasts less than 100 milliseconds and cannot be triggered by manual request. After a phase A initialization, the program is restarted from a fixed point and all maintenance activity is aborted. While no calls are disconnected, the delay of certain critical input-output tasks may result in problems which are later detected and cleared by call processing and audit programs; this migbt result in the mishandling of a few calls.

6.2.2 Phase B initialization

Phase B is the second level of initialization. During this pbase all but operator-to-subscriber calls are disconnected. Phase B can be requested manually or automatically, and lasts from four to eight seconds. This short duration is made possible because almost all peripheral circuits are electronic and, therefore, can be initialized at program speeds. For example, a call in the network can be disconnected in tens of microseconds.

When a phase B initialization is executed after a phase A has failed to recover the system, the most likely problem is mutilation of call

store data. Consequently, semipermanent areas of call store are initialized during phase B by the audit programs (Section 5.4) using a disc file backup record of nongeneric data. Several rapid checks of file operation, data validity, and plant changes in progress are performed to minimize the possibility of using incorrect data. Using the results of these checks, a preferred file is selected. If plant changes were in progress they are removed from the system. If neither file can be used, the semipermanent data previously in call store remain there.

All transient areas of call store are zeroed during phase B. Some call records are re-established when operator-to-subscriber connections are identified. This is done by reading the equipment numbers of connected trunks from the connector and scanner time slot memory. These are compared with the semipermanent trunk data which define the types of trunks assigned at each network equipment location. Transient call records are then re-established for all calls which are identified as operator to subscriber. Transient areas of call store are also initialized to reflect the service states of peripheral circuit and operator positions.

The order of activities during phase B is critical. For example, subscriber-to-announcement calls must be disconnected quickly to prevent improperly repeating the current announcement phrase or digit. In summary, the sequence of actions during phase B is:

(*i*) initialize peripheral circuits;
(*ii*) disconnect subscriber-to-announcement calls;
(*iii*) zero transient areas of call store;
(*iv*) initialize semipermanent areas of call store;
(*v*) disconnect all but operator-to-subscriber calls; and
(*vi*) initialize state data for peripheral circuits and operator positions.

The main program loop is then entered at a fixed point (at the beginning of the call processing monitor).

6.2.3 Phase C initialization

Phase C is the highest level of initialization. During this phase, all calls are disconnected; otherwise, it is much like phase B. However, phase C is slightly shorter in duration than phase B since subscriber-to-operator calls need not be identified and maintained.

The primary differences between phases B and C result from the two options which may be invoked when phase C is requested manually

from the emergency action panel. If either option is requested, phase C is entered even when no recent initialization has occurred.

The *stable* option causes stable data areas of call store (Section 3.1) to be zeroed. This results in loss of records such as time of day and plant and traffic measurements. No automatic initialization of stable data is provided because troubles in stable data are unlikely to result in improper call processing actions, and because such data cannot be automatically re-established. Since the initialization is requested manually, the requestor can simply re-enter the time of day via a teletypewriter following the initialization.

The *tape* initialization option is used when the integrity of both file backup records of nongeneric data is in doubt and the records must be restored from a backup punched paper tape. In this case, following initialization, the system is placed in a *tape mode* of operation in which no call processing is performed (Section 7.2).

Entry of the punched paper tape at a teletypewriter typically requires 10 to 20 minutes. After this period, normal operation is resumed. Tape mode is also used for initial system testing, since many maintenance and administrative functions can be executed without reference to nongeneric data. This mode also permits initial disc file loading of such data.

A summary of the initialization phase structure is shown in Table VII.

VII. ADMINISTRATIVE FUNCTIONS

7.1 Plant and traffic measurements

In administering an AIS, system facilities must be adequately engineered and maintained if satisfactory service is to be provided. *Plant measurements* are made which reflect the health of all system components by recording counts of system activities, failures, and maintenance states. These measurements are automatically printed at the maintenance teletypewriters and reset once every 24 hours.

In addition, facilities are provided at the AIC for recording *traffic measurements* needed for operator force adjustment and traffic engineering and equipment administration. These measurements consist of *peg counts* which are incremented each time an event occurs and *usage counts* which are incremented every 10 seconds. The counts for each type are printed half hourly at a dedicated traffic data receiving teletypewriter. Those counts associated with operator force adjustment are printed additionally at a dedicated traffic force adjustment teletypewriter.

Table VII — Summary of initialization phase structure

Phase (And Options)	Request Sources	Approximate Execution Time (Seconds)	Effect On Call Processing
A	Automatic	0.1	Very little
B	Automatic or Manual	4.0–8.0	All but caller-to-operator calls disconnected
C	Automatic or Manual	4.0–8.0	All calls disconnected
C Stable	Manual	4.0–8.0	All calls disconnected
C Tape	Manual	0.5	All calls disconnected; call processing inhibited

7.2 Administration of nongeneric data

Prior to placing an AIC in service, nongeneric data describing the AIC are punched manually on a paper tape and are placed on one track of each of the two disc files and in call store hy reading in this tape from a teletypewriter while in tape mode following a phase C tape initialization (Section 6.2.3). Thus, hackup copies of the nongeneric data are maintained on the files and also on punched paper tape.

When changes are made in equipment or options, the associated nongeneric data changes are made online hy a telephone company employee via a teletypewriter. In performing some of these changes, such as adding or deleting trunks, coordinated changes to the physical equipment may he required. Plant changes are performed hy entering a series of teletypewriter messages. As new data are received, the corresponding changes are made in the hackup data on one file.

Due to the sensitivity of the system to errors in the nongeneric data, the program makes checks for data completeness and reasonahleness as the new data are heing received. If the data are found to he incorrect, the request is rejected and a teletypewriter message is printed containing a code which indicates the reason for rejection.

When all plant changes have heen entered, a message indicating completion signals the audit programs (Section 5.2.2) to *regenerate* the nongeneric data in call store; this results in making the new data operational. Following regeneration, a period of testing ensues to insure proper system operation with these data. If a trouhle condition is detected, the original data can he restored from the unmodified file on manual request. After the new data are found to he acceptahle, the

telephone company employee requests the program to punch a backup paper tape which incorporates the new data. During this process, the program provides a printed record of the nongeneric data in the same format as it appears on the tape; thus, a listing of the tape for use as an office record is available. Finally, a request is made to read the nongeneric data from the modified file and write it onto the other file, bringing the two backup records into agreement.

VIII. SUMMARY

The foregoing discussion has provided the organization and structure of the operational programs, an explanation of the various functional tasks performed by the operational programs, and a detailed description of the call processing functions to show the manner in which autonomous circuits and software interact to process intercept calls. Novel features of the AIS have been emphasized: a single generic program, use of autonomous circuits, use of both hardware and software as the basis for audit techniques, and the methods of storing and administering nongeneric data. The authors have attempted to provide insight into the techniques and considerations used in the development of operational programs.

REFERENCES

1. C. J. Byrne, W. A. Winckelmann, and R. M. Wolfe, "Automatic Intercept System: Organization and Objectives," B.S.T.J., this issue, pp. 1–18.
2. J. A. Harr, Mrs. E. S. Hoover, and R. B. Smith, "Organization of the No. 1 ESS Stored Program," B.S.T.J., *43*, No. 5 (September 1964), pp. 1923–1959.
3. P. J. Brendel, W. K. Comella, R. N. Markson, P. J. Moylan, and J. Orost, "Automatic Intercept System: Peripheral Circuits," B.S.T.J., this issue, pp. 71–106.
4. J. W. Hopkins, P. D. Hunter, R. E. Machol, J. J. DiSalvo, and R. J. Piereth, "Automatic Intercept System: File Subsystem," B.S.T.J., this issue, pp. 107–132.
5. J. H. Carran, K. E. Greisen, W. G. Hall, and D. J. Wells, "Automatic Intercept System: Administering the Intercept Data Base," B.S.T.J., this issue, pp. 133–153.
6. T. E. Browne, T. M. Quinn, W. N. Toy, and J. E. Yates, "[No. 2 ESS] Control Unit System," B.S.T.J., *48*, No. 8 (October 1969), pp. 2619–2668.
7. M. E. Barton, N. M. Haller, and G. W. Ricker, "[No. 2 ESS] Service Programs," B.S.T.J., *48*, No. 8 (October 1969), pp. 2865–2896.
8. H. J. Beuscher, G. E. Fessler, D. W. Huffman, P. J. Kennedy, and E. Nussbaum, "[No. 2 ESS] Administration and Maintenance Plan," B.S.T.J., *48*, No. 8 (October 1969), pp. 2765–2815.